Data Discovery and Collection Architecture for Distributed IoT

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering

By

Rishabh Garg (141215)

Sachin Slathia (141230)

Under the supervision of

Mr. Arvind Kumar

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234,

Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled "**Data Discovery and Collection Architecture For Distributed IoT**" in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2017 to May 2018 under the supervision of **Mr. Arvind Kumar,** Assistant Professor (Grade-II), Computer Science & Engineering and Information Technology.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sachin Slathia (141230)

Rishabh Garg (141215)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Arvind Kumar
Assistant Professor (Grade-II)
Computer Science & Engineering and Information Technology
Dated:

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards **Mr. Arvind Kumar**, my re-search guide, for his patient guidance, enthusiastic encouragement and useful critiques on this research work. Without his invaluable guidance, this work would never have been a successful one.

The in-time facilities provided by the Computer Science department throughout the project development are also equally acknowledgeable.

Last but not least, my sincere thanks to all my teachers and my friends who directly or indirectly helped me to learn new technologies and complete my work successfully.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this report we have tried to find an optimal algorithm for Data Discovery for Distributed IoT and also extended this technique for Heterogeneous System using various techniques like **Edge Mesh, K -Centre** which is mostly used today in real world problems in networks. This will reduce Traffic in Wireless Networks. The responsibility of a single base station (which is easily hack able and If fails then whole devices connected to it will not work properly) is to distribute work among various base Station which are kept at optimal distance so that performance of wireless network devices increases.

We have also discussed various fault tolerance techniques so that if base station gets shut down or get destroyed then how will we recover data from base station so that devices connected to that base station does not get affected.

We have discussed various research papers we have read so far and conclusions and techniques we have taken from these papers and results we are trying to get using our algorithms and how can this project can be used in real time applications like shifting data from cloud to nodes. This project can be used in various areas like smart homes, smart city.

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

1.1.1 What is Data Discovery?

Data Discovery is a business client arranged process. It is the process of gathering information from number of databases and merging it into a source that can be easily managed.

1.1.2 What is Collection Architecture?

It implies design of accumulation of various detecting hubs with base stations send such that effectively distinguishes hubs that can answer to client demands.

1.1.3 What is Distributed IOT?

In IOT, there are billions of brilliant devices, for example, Sensors, Smart Phones, Smart Vehicles, and are associated with each other to in exchanging messages, queries in order to fulfill human needs.

In Real industry, Data is so tremendous – Infeasible and wasteful to deal with cloud administrations due to processing and correspondence assets. Circulated Computing is a fundamental procedure for IOT to off-stack the calculation from the servers.

1.1.4 What is a wireless sensor network?

It is a type of remote system that contains a large amount of self-coordinated, low powered devices called sensor networks. These systems certainly cover an enormous number of spatially distributed, almost battery powered, installed devices that are set up to gather messages process and exchange messages to the administrators.

1.1.5 Edge Computing (Fog Computing)

This is nearly an indistinguishable approach from what we have done in Distributed approach. All the calculation assignments and information are shared utilizing a work system of edge gadgets. Edge Mesh has numerous points of interest, including disseminated handling, low inactivity to non-critical failure, better adaptability, better security and protection.

Cisco presented this idea by conveying calculation from the cloud to the system

1.2 PROBLEM STATEMENT

1.2.1 Why Data Discovery and Collection Architecture is required?

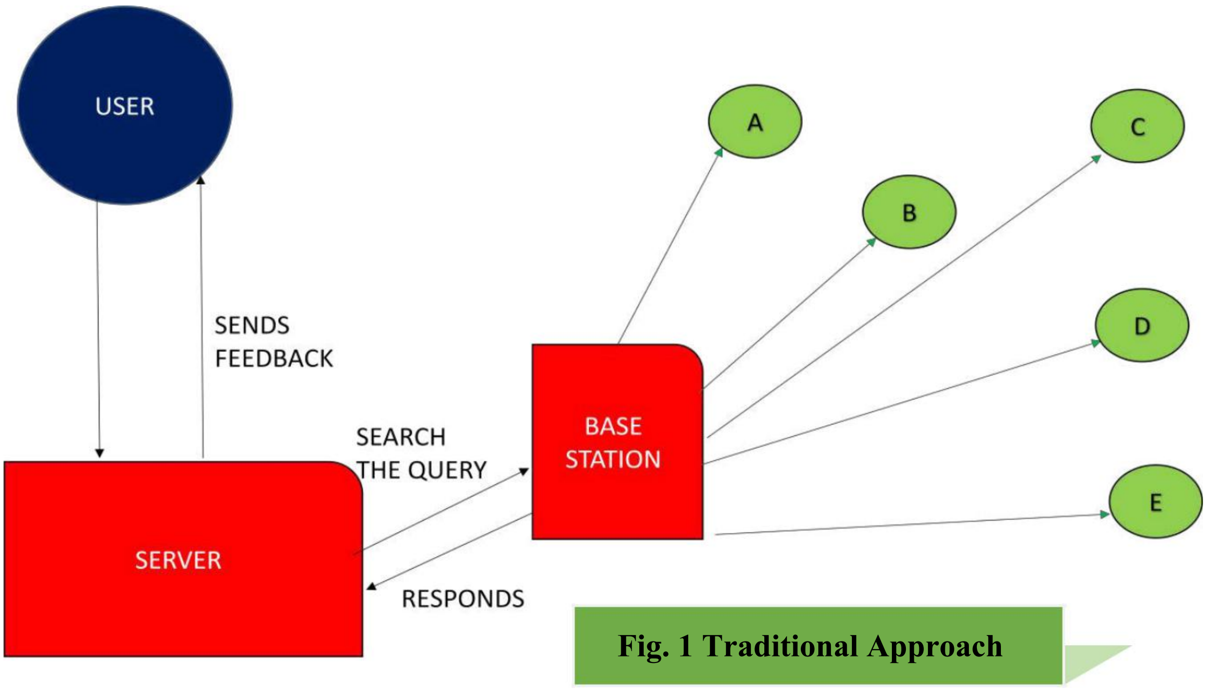


Fig. 1 Traditional Approach

This is the Traditional Approach. In this we make one base station which is associated with each hub. On the off chance that client needs to look through the question in server, at that point server looks through that inquiry in base station, at that point base station seeks which hub is associated with that inquiry. At that point information from each sensor hub reacts to base station to a server which at that point forms the information and transfers answers to the client. This expects system to be furnished with a hub prepared to do long range correspondence. Clients at that point recover information from the primary server utilizing the long-run correspondence. This is brought together approach that we took after.

In this If Base station fails, and then we can’t receive any data from nodes.

Network Traffic, Execution time is also the limitations of this approach.

So, we move towards Distributed Approach

Also discussed various fault tolerance techniques so that if base station gets shut down or get destroyed then how will we recover data from base station so that devices connected to that base station does not get affected.

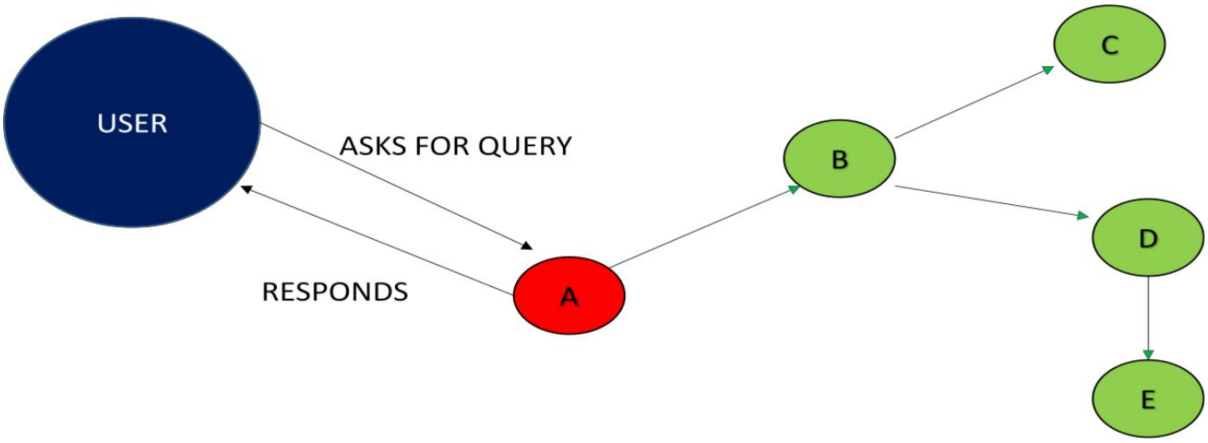


Fig. 2 Distributed Approach

In this approach, we have used distributed approach. Let's explain with example – If we have different sensing nodes in JUIT, then each department has its own Base station like for Library it has its own base station and that base station shares information with different sensor nodes which are only present in library. Now if user searches any query related to library sensing nodes and it easily collects the data from every sensor node which is involved in that query, send them through base-station to a server which then processes the data and relays answers to the user.

1.3 OBJECTIVES

We have confronted a few issues while we will apply this approach. Discoveries and Problems which we are recognized:-

1. Implement for heterogeneous system.

2. Which sensor node to make base station? How to decide to make it?

3. Number of base station in a network.

4. Reduce computational power of each node.

5. Fault Tolerance in WSN applications. To solve problem of fault tolerance. If some fault occurs how will our system responds.

1.4 METHODOLOGY

1.4.1 What are we going to do?

- Distributed algorithm for Heterogeneous System
- To identify and solve problem of Fault tolerance.
- Decide which node to make Base Station to reduce traffic in networks.
- How many Base Station we require in a network

1.4.2 How we going to do?

- Use k-center algorithm to decide which node to make Base Station to reduce traffic in networks and also how many Base Station we require in a network.

- Use Distributed Approach.

- Deployment of node to decide where to make Base Station

- **Fault Detection**
  - Centralized Approach
    1. Sympathy
    2. Secure Locations
  - Distributed Approach
    1. Node Self Detection
    2. Clustering Approach

- Genetic Algorithm proves to be best in dealing with problem of fault tolerance.

1.4.3 Tools and Technology used:-

1. C - programming language

2. Java

3. Cooja Simulator:-A Java platform in which we can deploy various IoT devices and implement our algorithm on these nodes also known devices in real life. It is useful as in real life we cannot have millions of devices with us so we use this devices.

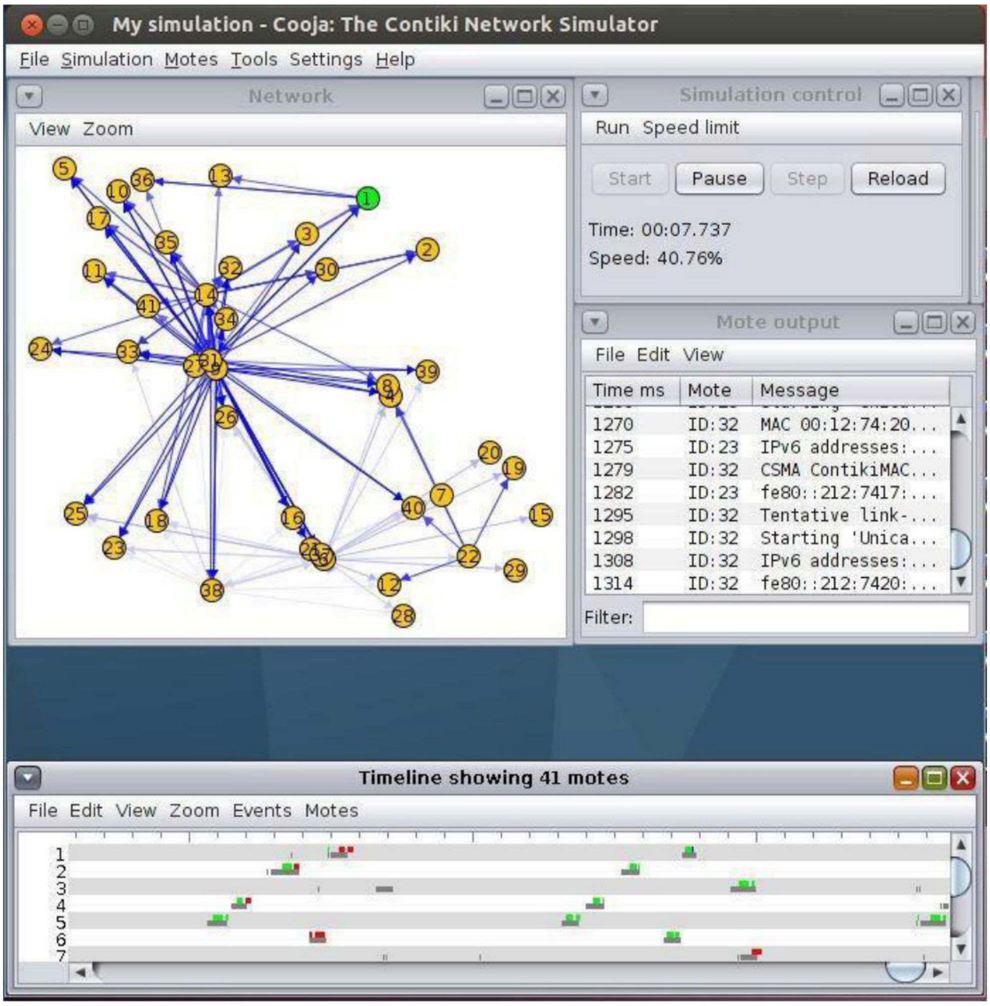


Fig. 3 Cooja: The Contiki Network Simulator

1.5 ORGANIZATION OF THE REPORT

In Chapter 2:- We have discussed various research papers we have studied that includes

- Distributed Data discovery and collection architecture for distributed IoT
- Optimal processing node discovery algorithm for distributed computing in IoT
- Edge Mesh: A New Paradigm to Enable Distributed Intelligence in IoT

In Chapter 3:- We have discussed various algorithms that are used in our project and how we will solve our problem using

- K-Centre and its limitations
- Matrix approach and its limitations

In Chapter 4:- Results and graphs

In Chapter 5:- Conclusions

CHAPTER 2
LITERATURE REVIEW

2.1 Distributed Data Discovery and Collection architecture for distributed IoT [1]:-

2018 came with drastic increase in WLSS devices. They are used everywhere and today each user is using a device. Every researcher in field of IoT is working on how to decrease traffic i n networks and DEVICE. Computational Power=LOW.

Dragon uses approach known as distributed approach which improves traffic in networks and DEVICE. Data=COMPUTATIONAL POWER. The device with low power is given data so that it does not go to failure and hence its lifetime increases and computational power of each node is decreased.

Example:-

Suppose there is a street in Noida which is full of restaurants. Now street owners decided to have a network in which a user can send his/her query and base node can implement the query and send the results back. Customers can send request to find an open restaurant with a vacancy.

2.1.1 TRADITIONAL APPROACH:-

1. Collect data from every restaurant periodically.
2. Send DATA (Base Node ->Server) which processes the given query and then reply to user.
3. Every device should be capable with long range communication.
4. Users will communicate with main server and will get answer for its query

2.1.2 Problem with this approach:-

Owner of the every restaurant may not pay mobile and server fees, and still wants to provide its services to users.

SOLUTION:-

Every node must:-

- Identify a number of sensor devices which can implement the request.
- Collect INFORMATION from EVERY DEVICE and REQUEST FROM REMAINING DEVICES.

Dragon solves above two of the problems by allowing any device to search in whole network the devices with given static requirements.

1. It also uses routing protocol which helps in sending messages to the devices.
2. It also uses Distributed data table which helps in storing data in distributed way.

2.1.3 Algorithms Used in This papers:-

1. Routing Table Discovery
2. Distributed Data Table
3. Static Attribute Propagation

2.1.3.1 Routing Table Discovery

Distributed technique consisting of RT for every device.
RT has A) Destination, B) Next hop, and C) Distance.

Algorithm:-

1. Every device has its own local RT in which it creates a record for every node and sends a discovery request to all devices.
2. Routing table has A) list of destination, B) Distance pairs.
3. After receiving discovery request UPDATE RT.
4. IF(RECORD NOT PRESENT)
5. Create new record.
6. IF(RECORD PRESENT && RECEIVED DISTANCE<ALREADY PRESENT) THEN
7. Update Distance And Save Next Hop.
8. Broadcast to all nodes.
9. If(FAILURE OCCURES) THEN
10. FALURE RECOVERY PROCESS=ACTIVATE.
11. Set NEXT HOP=UNREACHABLE for that node.
12. MESSAGES=CLOSER NODE.
13. ELSE
14. RECORD is marked as update

2.1.3.2 Distributed Data Table

Every node in network has A) Id, B) Type, and C) Room Id.
A table consists of ST where COLUMN=ATTRIBUTE, ROW=NODE.

Algorithm:-

1. NODE. Information=static attributes.
2. WLSS. Memory=LOW THEN
3. USE DDT.
4. Divide whole network in z parts
5. NODE. Memory=HIGH THEN
6. Store (z-x) part.
7. Repeat for every device.

HOW QUERY PROCESSING TAKES PLACE

A node receives a query for finding device more than 10 capacity. It looks in its local Distributed data Table. If (Not Found ==TRUE) forward to z-1 devices. Devices reply after searching in their local DDT.

Challenge with this approach is that's how to assign parts of a node in each device. So that network has to send minimum number of messages.

2.1.3.3 Static Attributes Propagation

Function receive

1. Make a Function "receive" and it has static attributes and sender id.
2. If buffer is empty then add into buffer.
3. Store the current time and random delay into send function of Buffer.
4. Get the list of neighbor and stores into receive function of buffer.
5. Now, if the Distributed Data Table of this buffer matches with Id then
6. Insert into Distributed Data Table of buffer.

Function send

1. Make a function "send" and it has buffer attributes.
2. If the received function of buffer is not empty then broadcast the buffer attributes.
3. Remove the buffer attributes from Buffer.

In this approach, we choose a random node and this random node collects the Distributed Data table for all the neighbors and waits for some time. According to these Distributed Data Tables, this random node selects which node executes next.

2.2 Optimal Processing Node Discovery Algorithm for Distributed Computing in IoT [2]

2018 came with huge increase in internet users and hence devices. These devices periodically send lot of data stream, where many devices are to be processed and stored.

We are sending whole data to cloud.

Disadvantage of cloud is it increases data processing latency which can sometimes cause unnecessary energy inefficiencies, including unnecessary data transmission.

In Distributed platform all operations takes place on the network. It reduces traffic in network.

Algorithm:-

Each device interacts with user takes the query from user find the desired nodes and respond to query so that computational power of each device is reduced.

2.2.1 New Platform for efficient in–network Data Stream Processing

There are 2 algorithms for it:-

1. **PND (Process Node Discovery):-** This algorithm is used when there are continuous queries in network. It uses distributed approach to find optimal processing node.
2. **QTB (Query Tuple Buffering):-** This algorithm is used just after PND to reduce traffic in network.

There are requirements for query processing:-

1. Query will act as universal sometimes requires information from every device in the network.
2. How much time will it take to process a query? There is a query known as Snapshot query which is executed once and hence for query execution optimization there is no space.
3. To counter the query the query is executed many times.

2.2.1.1 Processing Node Discovery Algorithm

This algorithm plays important role in finding the devices which fulfill the requirements of the given query.

In this algorithm we have given user access to communicate with any node in network and every device can respond to queries by user.

When a user sends a query the 1st node to receive that query is called starting node (initiator). The initiator uses DRAGON to see which nodes are going to participate to solve the query.

The initiator sees DDT to find devices. The nodes which participate in solving the query are called sources.

Objective of Algorithm

Objective of this algorithm is to find a node whose cost to all source nodes is minimized. The network receives query from user and then sees which node has minimum cost to perform this operation. The base node computes the cost for query and gathers cost from all neighbors.

If cost of a neighbor is low then make it the base node else from its neighbors calculate the node with minimum cost and make it base node and all queries will be received and executed by it.

2.2.1.2 Query Tuple Buffering Optimization

As base node receives query it forward the request to processing nodes. The messages pass by a common node and this node merge two or more queries into single message instead of forwarding each message separately .This node is called merging node. Thus traffic is reduced in network.

The maximum difference between arrive of 1st message and last message in a single query is called as Maximum delay. If all messages don't arrive in this span of time then all messages are sending back to the base node which is low in computational power.

.

2.3 Edge Mesh: A New Paradigm to Enable Distributed Intelligence in Internet of Things [3]

Now a days, Technology shifts from Cloud Computing to Edge Computing. This edge computing is used by CISCO. They transfer their data from a centralized server to Distributed network. In that they make some base stations according to their energy consumption and energy utilization of a node. These base stations are connected with the end devices. Enhancement in technology increases the computational capabilities of the embedded devices. Nodes (IoT devices) are very low powered so we cannot utilize these devices for any computational activity. We can distribute the computational activities to edge devices from the centralized network by Edge mesh. Edge mesh gives numerous advantages – better security, better scalability, fault tolerance, distributed processing and privacy. It also provides real time processing. Main problem in above surveys that we have to consider only homogeneous network i.e. all nodes have similar configurations. But, in real life environment we have to consider heterogeneous network i.e. nodes have different configurations because in real life almost every devices have different configurations. So, in edge mesh network we can use heterogeneous end devices because in this we only distribute the base stations according to their configuration.

A large portion of the IoT applications have four principle segments i.e. Communication, Sensing, Computation and Actuation. End devices in edge mesh network have sensors which are used for sensing the environment. Gateways and routers are used for communication.

Now, Computation is generally done at base stations. They compute the information and send it to some end devices (Actuators). Computation is a critical piece of IoT as it prompts information creation which can be used to give good services.

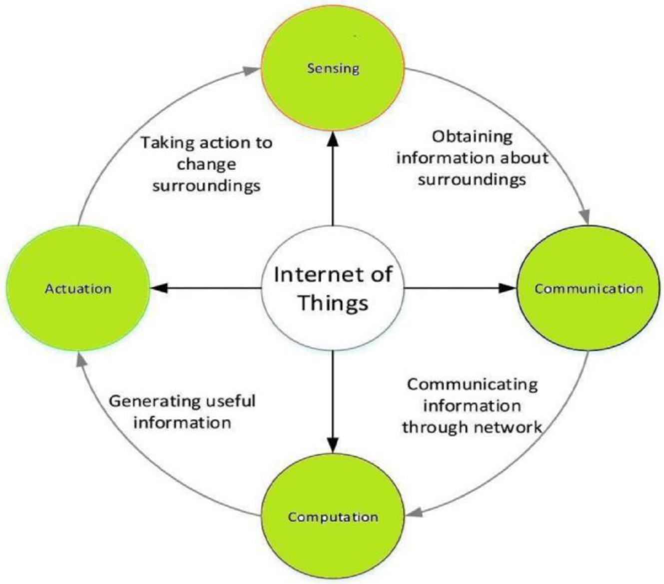


Fig. 4 IoT devices

2.3.1 Task Allocation in Edge Mesh

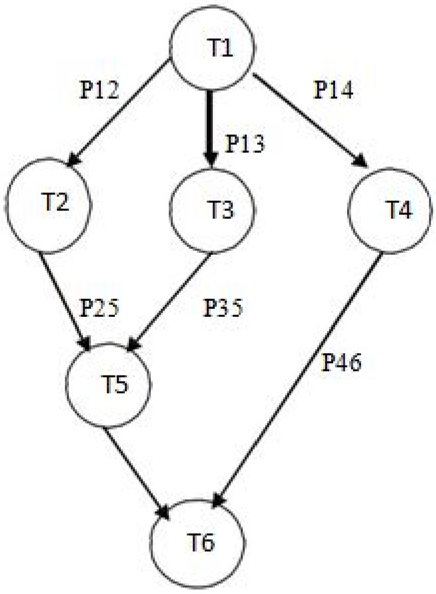


Fig. 5 Task allocation graph

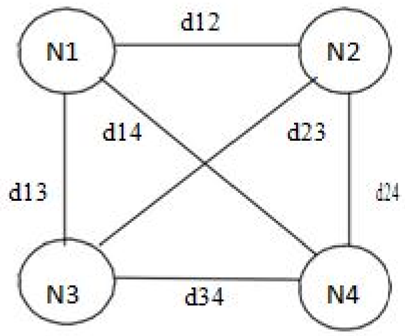


Fig. 6 Communication network

T_i -> i^{th} task

P_{ij} -> length of data, this data is transferred between t_i and t_j N -> Number of devices

d_{jk} -> distance between a_j and a_k

In this example, we have 6 tasks and 4 devices. Now, we have to give these tasks to edge devices. Here, one thing always keeps in mind that we have to minimize the total energy consumption .In edge mesh network, base stations are responsible for decision making and task allocation according to their configuration. End devices are deployed in any region so these can be far away from each other and end devices have sensing devices which sensed the data for different tasks.

2.3.2 Application

Falling-asleep problem

In this situation, it is envisioned that a man who is sitting on a sofa and sitting in front of Television. Now, if this man gradually falls asleep.

Now we want to change the environment accordingly so that we can save the energy and that man can get a pleasant sleep. So, for that we have to make the devices smarter like automatically Television should be switch off, darkening the lights, changing the aeration and cooling the system according to surrounding temperature, sofa becomes bed, etc.

For all these, all the devices are connected to internet so that instructions or data can be computed and processed. Now imagine these all devices are nodes and these are connected with each other. When you get falls asleep your body temperature changes so, end devices sensed this information and given it to base station which can computed and making some decisions. After that this base station distributes the task to other devices like darkening the lights, switching off the TV, sofa becomes the bed, changing the cooling system of room. These all are smart devices. These all require some algorithm that how to execute and processed the edge devices in edge mesh network. If this whole process runs on a centralized network then this would take more time and security threat as whole data would be shared at a centralized server. So, for this we distribute the processing load to different devices. This can give better results in terms of security, time consuming, energy consumption and cost.

2.4 Fault Tolerance [5]

- When a fault occurs in a network how will our network respond to that fault and what recovery techniques it will use
- Each sensor node is equipped with 2 components i.e. sensing unit and wireless transceiver. Sensing unit is utilized to take events and Wireless transceiver is utilized to change the captured events back to the base station called sink node.
- Sensor nodes work together with each other to perform tasks of data communication, sensing and processing.

2.4.1 Types of Failure in Wireless Sensor Networks

- Energy depletion– These devices have very limited energy and their cells can't be replaced or recharged, because of adverse environments.
- Hardware failure- A sensor node contains 2 segments: sensing unit and wireless transceiver.
- Communication link errors -The sensor nodes communication is affected by numerous elements, such as antenna angle, signal strength, obstacles and weather conditions, even if condition of the hardware is good.
- Malicious attack-It brings about low reliability of execution of sensor nodes.

2.4.2 Fault Detection

Centralized Approach

2.4.2.1 Sympathy

- Utilizing message flooding way, manage with data and current states from sensor nodes.
- Nodes periodically send current states back to a sink to identify failures and reason for that.
- These transmitted current states should be minimized.
- Information should be adequate at the sink to run the network properly. If information is not proper then the sink infers failure.
- Based on these current states, it recognizes which nodes have not conveyed adequate information and infers the reason of failures.

2.4.2.2 Secure Locations

- Work on Location- aware sensor networks.
- Select a protected way and keep away from insecure locations.
  1. Sink makes a message. And this message contains source location, destination location, authentication message.
  2. Encrypts this message with a key and communicates it.
  3. Neighbors who know its key will have the capacity to decrypt the request.
  4. Neighbor decrypts this request, and attaches its location with the message, again encrypts that message with a key and sends it to other neighbors.

2.4.2.3 Advantages and Disadvantages of Centralized Approaches

- The Centralized Approach is proficient and precise to distinguish the network faults in certain ways.
- It is not possible to periodically collect all the sensor states in a centralized network as we have a sensor network which has resource constrained.
- Central node is easily detecting all the fault detection but it is facing a large data traffic problem (single point).
- This will become costly in terms of large scale sensor network.

2.4.2.4Distributed Approach and Node Self Detection

- We have to use a circuit which acts as a sensing layer around a node. It is efficient to detect the condition of a node.
- It detects the physical faults by:
  1. Hardware interface has a sensing layer around the node.
  2. Software interface fetch the data from the sensors and transfer the data to the sink.

2.4.2.5 Distributed Approach and Clustering Approach

- It is the design for event directed Wireless Sensor Network.
- This mechanism reduces the energy consumption and data flow.

Installation

- Each node announces its energy and position to the operator situated in the network head.
- Operator forwards an ENERGY and LOCATION to the Controller.
- Controller constructs Wireless Sensor Network energy model and the topology map model.

Operation

- Whenever there is a state change, each node announces its energy and position to the operator and operator forwards to the controller.
- Controller reconstructs energy model and topology map model. Controller forwards GET function to get the node state.

2.4.3 Fault Recovery

- We can replicate or redundant the components to prevent the data failure.
- Whenever any node fails to provide the data, the base station knows which node gets failed. Then the data is sent to user from that redundant sensor node.

CHAPTER 3
ALGORITHMS DESIGN

3.1 Problem Description

Given n devices and distances between every device, select k devices to make base station in a region such that energy consumption of that device is minimize, energy utilization is maximize and distance of that device from other devices is minimum.

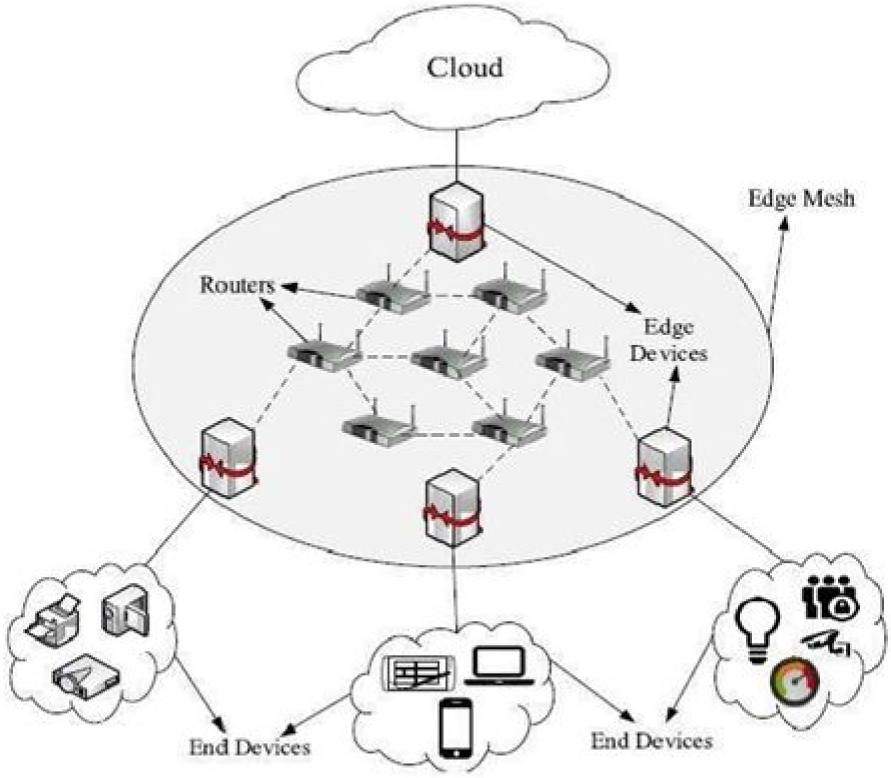


Fig. 7 Overview of Problem

3.2 APPROACHES

Approach 1

We will divide the given network into several base devices. Now these new base devices are responsible for devices in there corresponding regions to take the desired query and reply the result to these devices.

How will we decide which node to make base station so that cost in network reduces??

We will use greedy K-center algorithm to decide

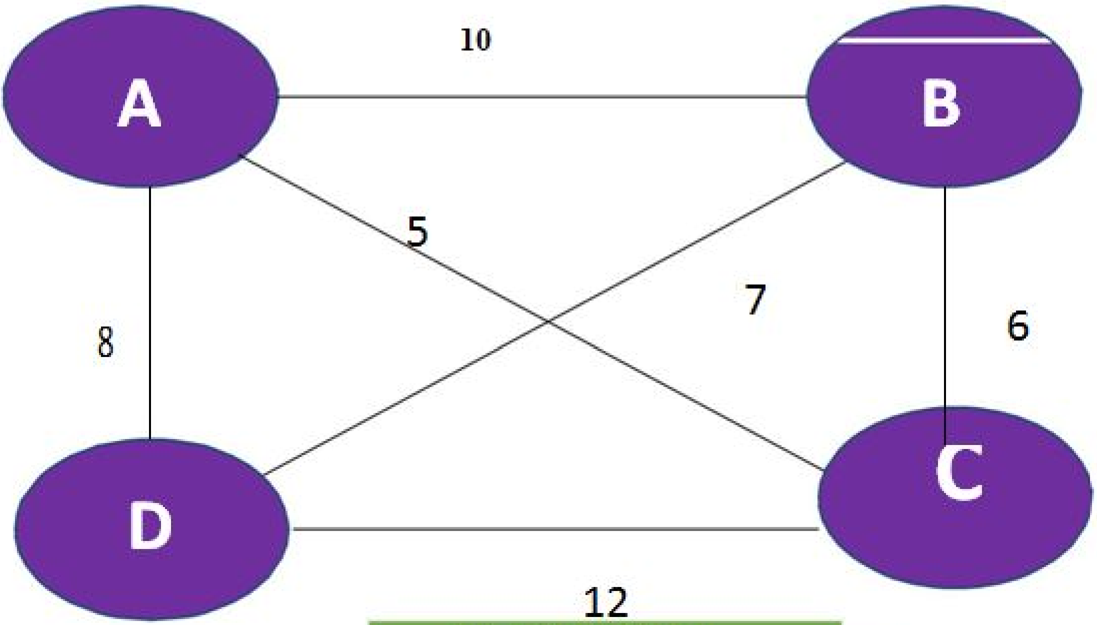


Fig. 8 K-CENTER

Suppose there are 4 devices (A, B, C, D) in a network and cost to every device is given

a) Randomly pick any node let it be B.

b) Pick next node which is farthest from above node that is A.

c) C and D are left.

Min [distance (D, B), distance (D, A)] = Min [7, 8] = 7 Min

[Distance (C, B), distance (C, A)] = Min [6, 5] = 5

After computing the above values, the Node D is picked as the Base node as every node has minimum distance from it and maximum energy utilization.

- We will use this algorithm in our project to find the nodes that we give responsibility to communicate with other IoT devices and we will also see on how much distance we have to keep this base node so that energy is minimized in our network and traffic is also reduced. Also how many minimum base nodes we need to have in a network so that each device can communicate with its nearest base station in most optimal way.

Limitations of First Approach

1. In K – center problem, initially we have to assume the number of Base Stations. After executing this problem we get to know actually how many base stations we have to choose. Like in our example we have taken n = 4 cities and for that initially we assume 2 ATMs should be placed for 4 cities so that distance between the cities and ATMs should be minimized.
2. This is an approximate algorithm. This algorithm can't always execute in polynomial time. This can be executing in exponential time also.

3.3 Approach 2

How to decide which node will act like Base Station?

Algorithm

1. Take an input - number of nodes.
2. Take an input – Cost of every adjacent node.
3. We use matrix to store cost of every device.
4. MEAN=TOTAL COST/NUMBER OF NODES.
5. If(MEAN<=COST OF NODE ROW WISE) THEN
6. Device will act as base station.

Let's take an example we have 5 nodes. They are connected with each other. There are costs to go from one node to another.

This is bidirectional network.

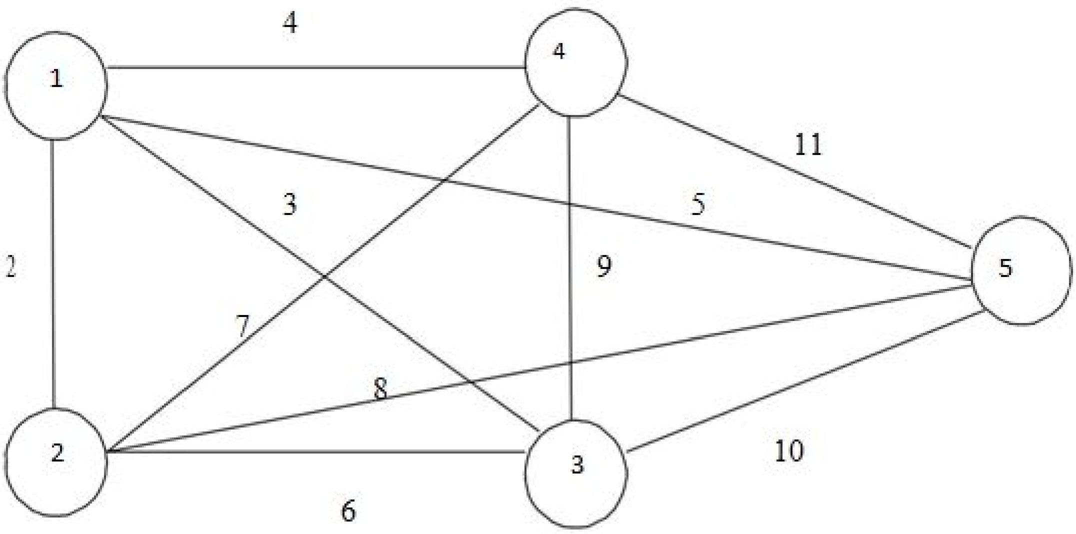


Fig. 9 Communication network

Let's say these nodes are deployed in real life environment. These nodes can be the mobile phones, or any device which can be connected to internet. Now if any user wants to send the query from any node to other node. Here we want to minimize the cost of travelling the query from one node to other so we have to make some Base stations for that. Now we see which nodes will become the base stations. So we make algorithm for that. We choose the node which has less cost with other nodes so that our query can be processed in minimum cost and time.

Here we make matrix for the same network. In this we store the costs of sending the query from one node to other nodes. In this network, if the cost of sending the query from node 1 to node 2 is 2 units then the cost of sending the query from node 2 to node 1 is also 2 units because this is bidirectional network.

Table 1 Cost Matrix

	1	2	3	4	5
1	0	2	3	4	5
2	2	0	6	7	8
3	3	6	0	9	10
4	4	7	9	0	11
5	5	8	10	11	0

Table 2 Row wise cost

Total	14	23	28	31	34

-

Mean = 130/5 = 26

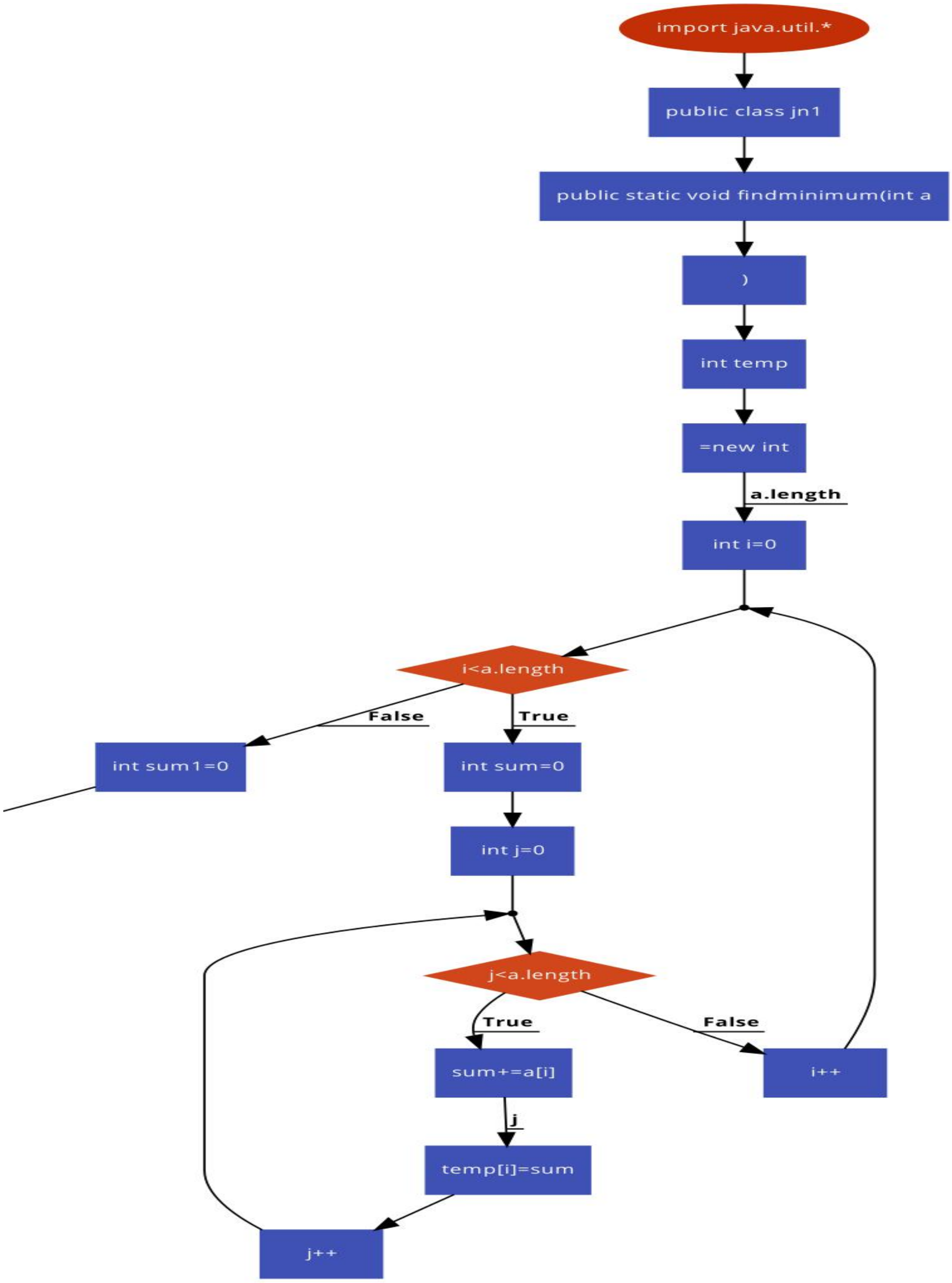
For this example, Mean is 26. Now we see which node has total cost lesser than this Mean cost. In this Node 1 and Node 2 have total cost lesser than Mean cost. So, these nodes become the base station.

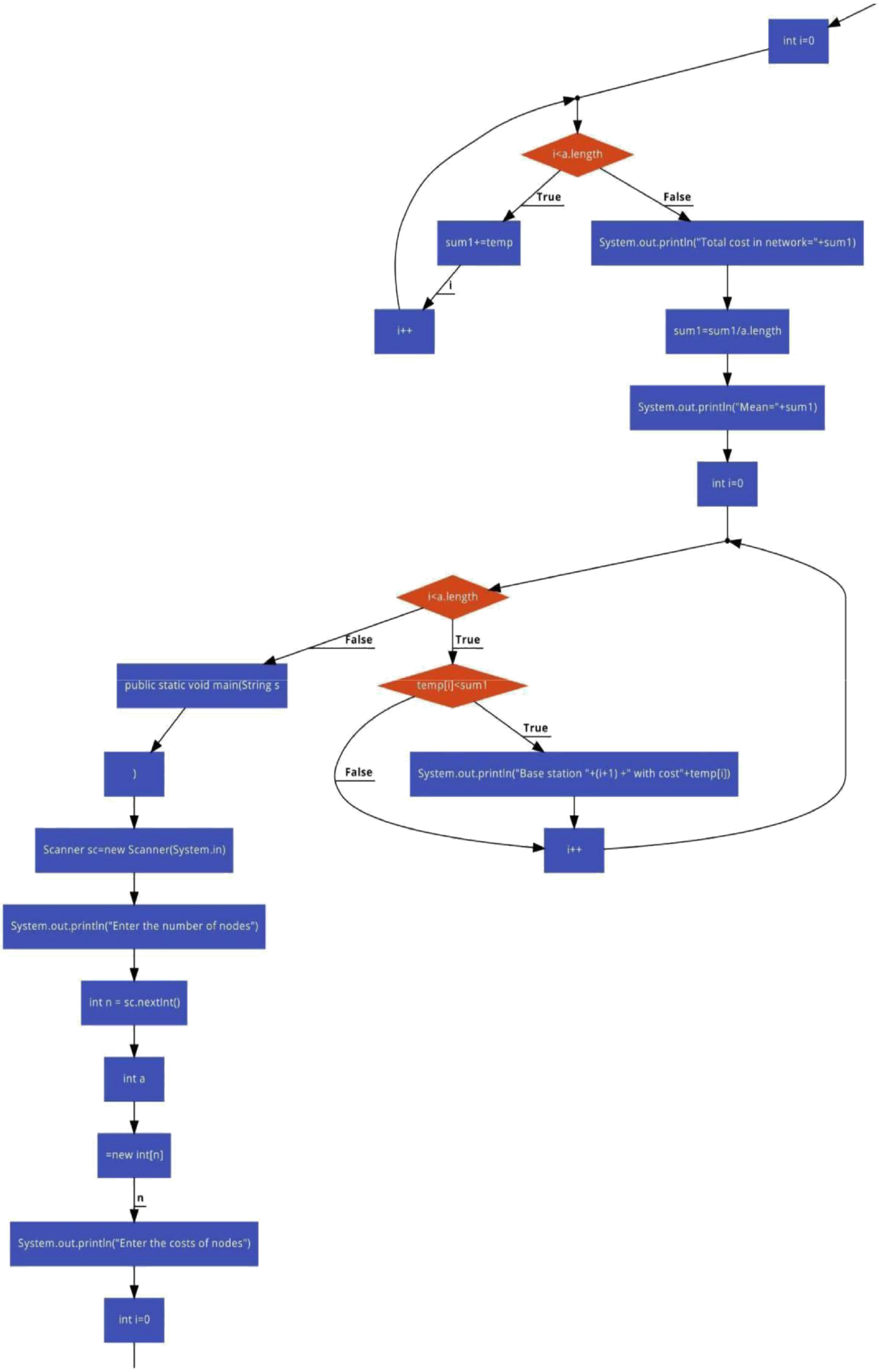
This means, if the user wants to send the query from any node then the node1 and node 2 become the base station. Then any query will be sending from any node to other via these base stations. This will reduce the cost and time.

Limitations of Second Approach

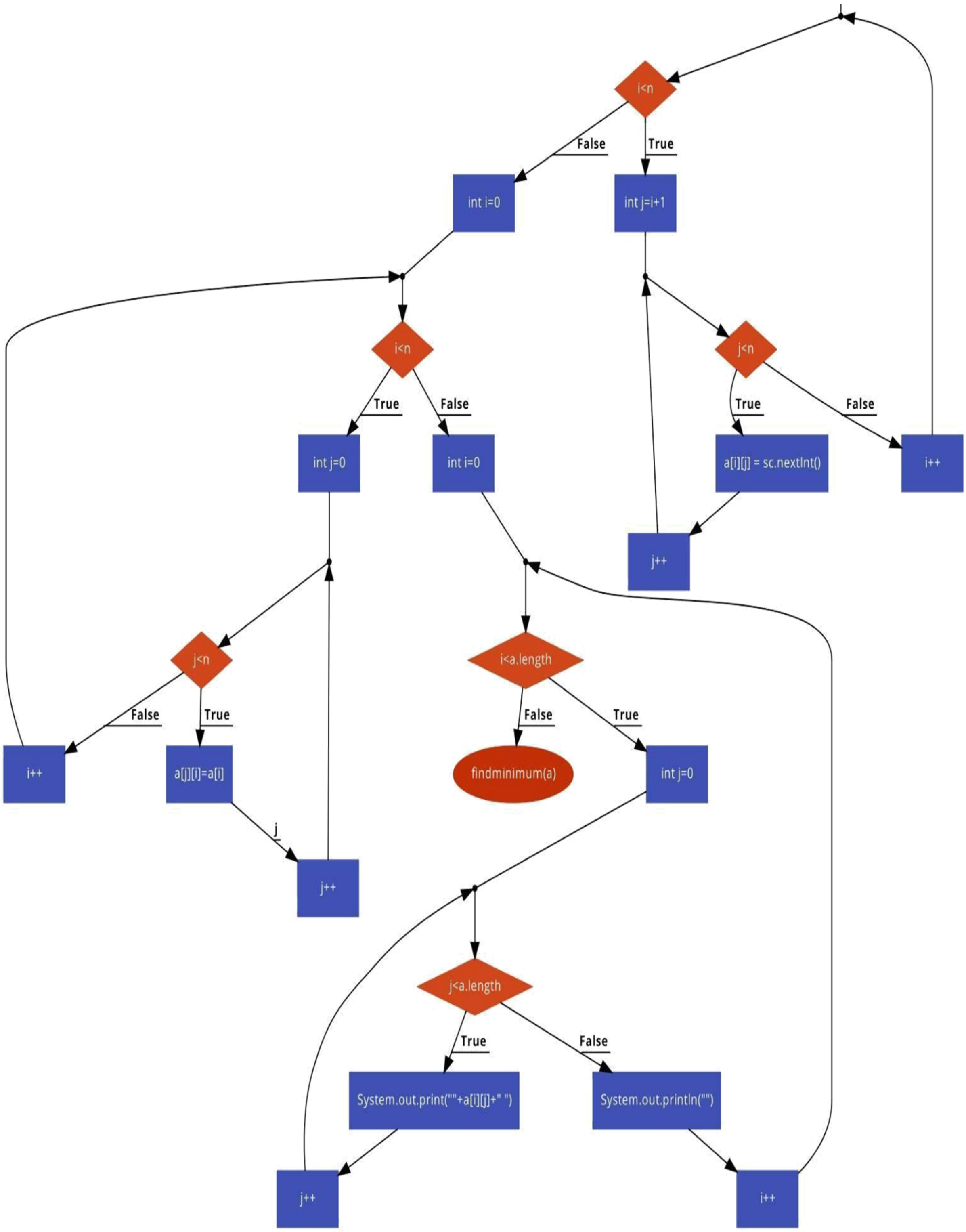
1. This takes more space because this makes a matrix for whole network like if we store the cost from node 1 to node 2 is 4 units so this matrix automatically stores the cost from node 2 to node 1(4 units).
2. In this algorithm, if the cost of network increases then the number of Base stations also increases. Now, the load on some base stations which has very low power increases.
3. Sometimes, some nodes are of low cost, low energy consumed but they are very low powered devices. If these nodes will become the Base stations then these can be easily failed.

3.4 Flow chart





int i=0
i<a.length
True
False
sum1+=temp
System.out.println("Total cost in network="+sum1)
i
i++
sum1=sum1/a.length
System.out.println("Mean="+sum1)
int i=0
i<a.length
False
True
public static void main(String s
temp[i]<sum1
True
)
False
System.out.println("Base station "+(i+1) +" with cost"+temp[i])
i++
Scanner sc=new Scanner(System.in)
System.out.println("Enter the number of nodes")
int n = sc.nextInt()
int a
=new int[n]
n
System.out.println("Enter the costs of nodes")
int i=0


i<n
False
True
int i=0
int j=i+1
i<n
j<n
True
False
True
False
int j=0
int i=0
a[i][j] = sc.nextInt()
i++
j++
j<n
i<a.length
False
True
False
True
i++
a[j][i]=a[i]
findminimum(a)
int j=0
j
j++
j<a.length
True
False
System.out.print(""+a[i][j]+" ")
System.out.println("")
j++
i++

We make a class jn1 in that class make a function i.e. find_minimum with a parameter (an array a[][]). Then we make a temporary array of same length size.

In that array we have to store the sum of row wise matrix. So, we have initialized a variable (i) for a loop and after check the condition i.e. i<a.length if true initialize the variable sum=0 and start nested loop statement to add the cost of every node and store it into temporary array else exit from the loop statement.

After that again loop starts for addition of all elements present in temporary array and store it into variable (sum). Then we divide the sum by number of nodes. Here we find mean.

Now starts the loop for checking the ‘if condition’ if costs which are stored in temporary array is less than mean they become the Base stations.

Then start the main function, in that store the number of nodes and respective costs with their neighbors and call our function (find_minimum).

CHAPTER 4

RESULTS

We came with the algorithm to find base stations and we have tested the algorithm for various nodes and found the corresponding base stations and have plotted a graph between number of nodes and number of base stations.

Table 3 Inputs

number of nodes	Number of base stations
2	0
3	1
'4	2
5	3
8	3
10	6

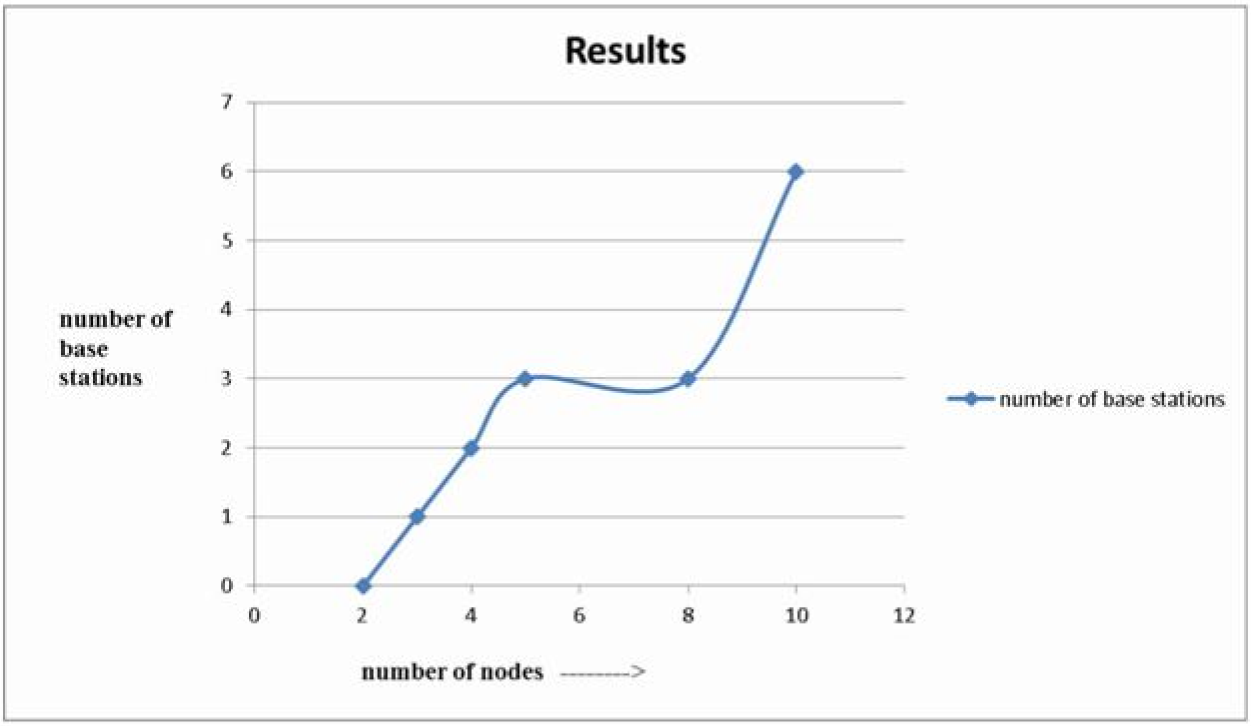


Fig. 11 Graph

The graph in Fig.10 is between number of nodes and number of base stations. We can see that as number of nodes increases and therefore cost in network increases and hence number of base nodes increases.

```
CPU Time: 0.23 sec(s), Memory: 32668 kilobyte(s)

Enter the number of nodes
10
Enter the costs of nodes
0 2 5 4 3 1 2 2 3 5
2 0 6 2 1 3 4 5 2 1
5 6 0 1 3 5 4 6 7 8
4 2 1 0 9 4 5 2 1 1
3 1 3 9 0 3 3 2 2 1
1 3 5 4 3 0 1 2 1 3
2 4 4 5 3 1 0 3 5 7
2 5 6 2 2 2 3 0 2 1
3 2 7 1 2 1 5 2 0 4
5 1 8 1 1 3 7 1 4 0
Total cost in network=294
Mean=29
Base station 1 with cost27
Base station 2 with cost26
Base station 5 with cost27
Base station 6 with cost23
Base station 8 with cost25
Base station 9 with cost27
```

Fig. 12 Snapshot of output window

Here we have entered the number of nodes and their corresponding costs and then cost of whole network is calculated and desired base stations are chosen which will do desired tasks.

CHAPTER 5

Conclusions

We have developed algorithm to find the base stations and we have successfully tested it on number of base stations and number of nodes.

In future we will be trying to deploy this algorithm in real world network where number of devices is more and traffic is high and see the desired results whether this technique can be deployed in real world or not.

We are also planning to deploy this algorithm in heterogeneous system.

List of References

[1] R. Kolcun and J. A. McCann, “Dragon: Data discovery and Collection architecture for distributed IoT,” in Internet of Things 2014 (IoT 2014), Cambridge, USA, Oct 2014.

[2] R. Kolcun, David Boyle and J. A. McCann, “Optimal Processing Node Discovery Algorithm for Distributed Computing in IoT”, in Internet of Things 2014 (IoT 2014), Cambridge, USA, Oct 2014.

[3] YUVRAJ SAHNI, JIANNONG CAO, (Fellow, IEEE), SHIGENG ZHANG, (Member, IEEE), AND LEI YANG, “Edge Mesh: A New Paradigm to Enable Distributed Intelligence in Internet of Things”, in Internet of Things 2017 (IoT 2017), Changsha, China.

[4] Jun Guo and Hamid Jafarkhani, “Sensor Deployment in Heterogeneous Wireless Sensor Networks”, in Internet of Things 2011 (IoT 2011).

[5]Savita Gupta, Sukhwinder Singh, Gholamreza Kakamanshadi,“A survey on fault tolerance techniques in Wireless Sensor Networks”, in Internet of Things January 2016, Noida.

[6] M. Kovatsch, M. Lanter, and Z. Shelby, “Californium: Scalable cloud services for the internet of things with coap,” in Internet of Things 2014 (IoT 2014), 2014.

[7] A. Uchechukwu, K. Li, and Y. Shen, “Energy consumption in cloud computing data centers,” International Journal of Cloud Computing and services science, vol. 3, no. 3, 2014.

[8] [48] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed Internet of Things,” Comput. Netw., vol. 57, no. 10, pp. 2266_22, 2013.

[9] W. Shi and S. Dustdar, “The promise of edge computing,” Computer, vol. 49, no. 5, pp. 78_81, 2016.

[10] OpenFog. De_nition of Fog Computing. Accessed on May 14, 2017. [Online].
Available: http://www.openfogconsortium.org/resources/ #definition-of-fog-computing

[11] S. Ray, Y. Jin, and A. Raychowdhury, "The changing computing paradigm with Internet of Things: A tutorial introduction," IEEE Des. Test, vol. 33, no. 2, pp. 76_96, Apr. 2016.

[12] J. Yang, H. Zhang, Y. Ling, C. Pan, andW. Sun, "Task allocation for wireless sensor network using modi_ed binary particle swarm optimization," IEEE Sensors J., vol. 14, no. 3, pp. 882_892, Mar. 2014.

[13] S. Feng, P. Setoodeh, and S. Haykin, "Smart home: Cognitive interactive people-centric Internet of Things," IEEE Commun. Mag., vol. 55, no. 2, pp. 34_39, Feb. 2017.

[14] http://www.geeksforgeeks.org/k-centers-problem-set-1-greedy-approximate-algorithm/